SNAP CIRCUIT – BASIC MOTOR CONTROL

WARNING

Make sure you pay close attention to the details, as failure to do so could result in your Raspberry Pi suffering damage and invalidating the warranty. We cannot be held responsible for any damage resulted with incorrectly following the guide.

DESCRIPTION

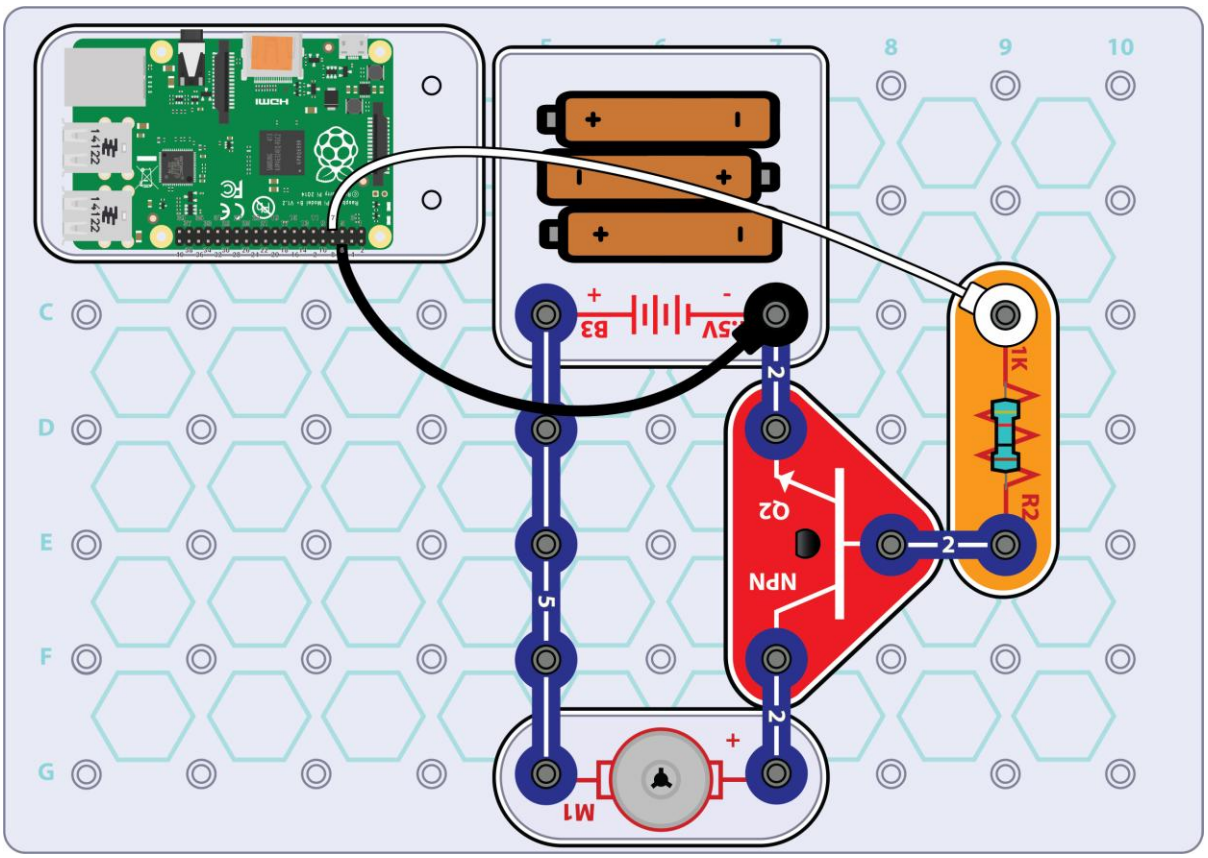
The Raspberry Pi is very good at talking to many different electrical components with the help of digital input and outputs, serial and with Bluetooth and Wi-Fi. Due to power and voltage limitations of the Raspberry Pi GPIO, additional components are needed to control high current parts like motors.

HIGH CURRENT

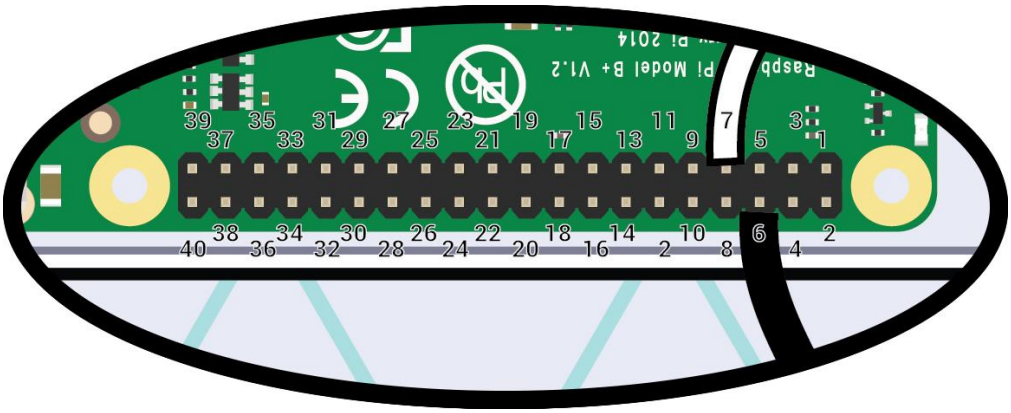
As the GPIO pins offer no protection to the Raspberry Pi SoC (system on a chip – The big black square in the middle) we must take extra precautions not to damage it. The complexity of the SoC results in that any damage inflicted would render the entire Raspberry Pi useless.

Transistors act like a 'digital switch', capable of turning on from another source (batteries in this case) so that the Raspberry Pi can control higher current or voltages with ease. Adding an intelligent controller to a motor or another output can allow remote/timed controls, interaction with games, receiving emails – Perfect for merging physical computing with software.

THE CIRCUIT



Firstly place the battery plate, motor, R1 and Q2 transistor on the base plate. These will go first as the bridges sit on top. If you would like the fan to spin off the motor, ensure the + marking on the motor is next to the transistor. Link up the components with the 5 snap bridge and the 2 snap bridges.



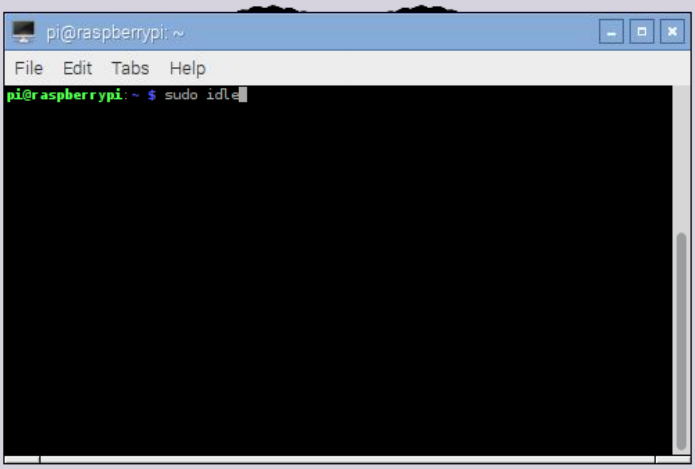
Now to connect the Snap to Pi wires between the Raspberry Pi and circuit.

Carefully count the number of pins on the Raspberry Pi before attaching any wires, be sure to double and triple check that the wire matches the correct pin!

When you're happy everything looks good, add the batteries. Hopefully nothing will happen.. Yet.

THE PYTHON CODE

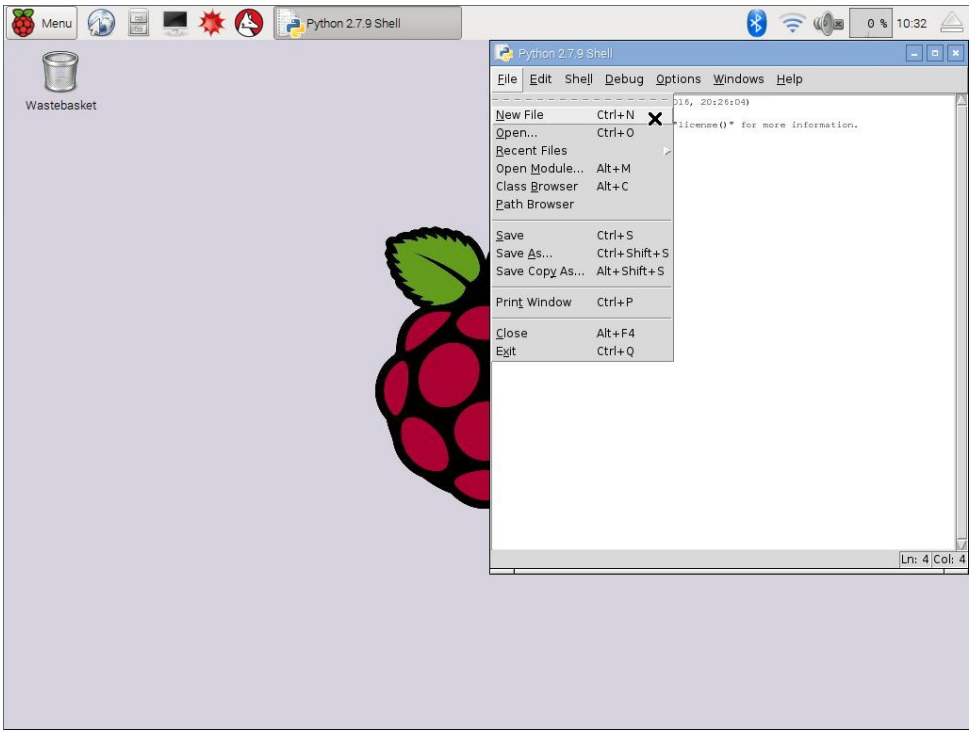
With the Raspberry Pi booted and showing the desktop, open Terminal by either clicking the Black computer screen icon on the task bar, or click Pi Menu (start button), followed by Accessories and then Terminal.

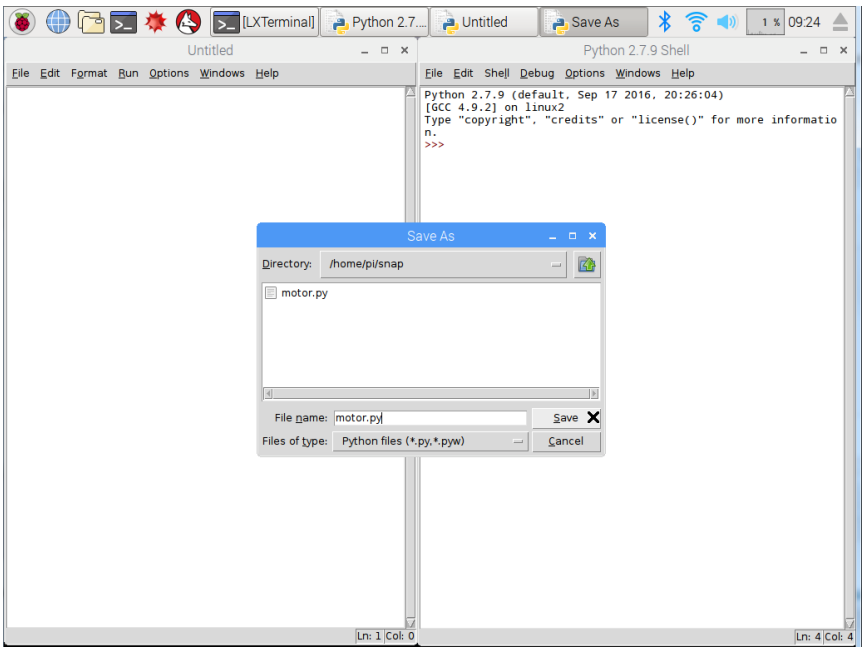


In this box type in sudo idle. This will open IDLE for typing our Python code with permission to control the GPIO pins.

Click File and then New File;

In the window that pops up, click *File*, *Save As* and give it a name to remember





Type in the following code in to this window, pay attention to capitals and spacing;

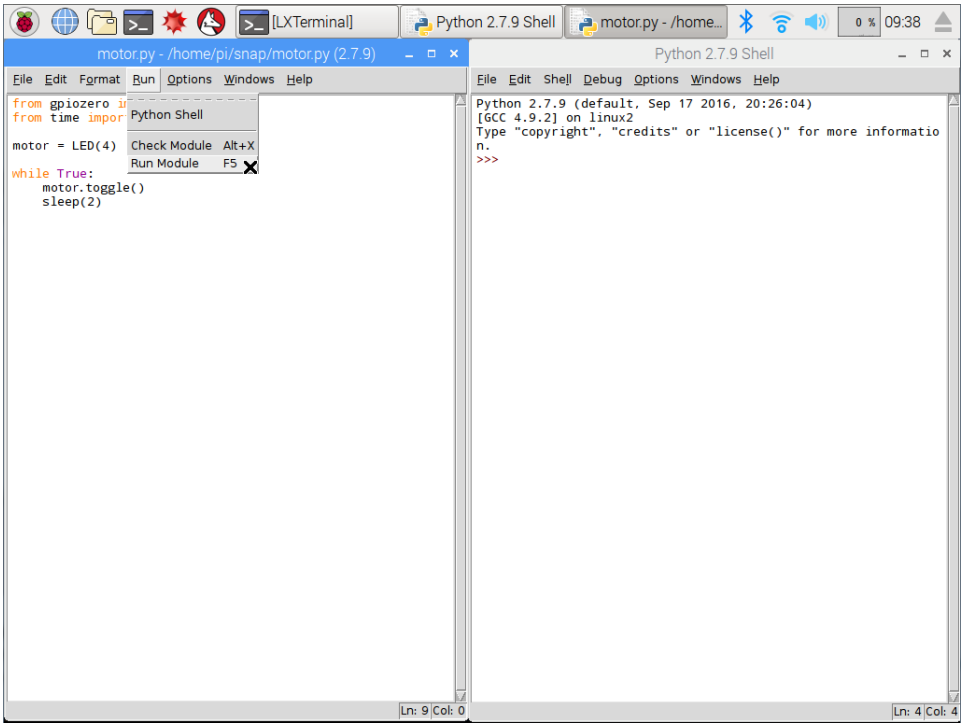
```
from gpiozero import LED
from time import sleep

motor = LED(4)

while True:
    motor.toggle()
    sleep(2)
```

Now double check the code to make sure no mistakes were made, otherwise the code might run and not give the expected results, or present you with an error.

When you are happy, click Run, and then *Run Module*.



Immediately the motor will start to spin for 2 seconds and then stop for a further 2 seconds before starting again. As this was created with a while True: loop, this will run forever until told to stop.

To actually stop the script, just hold CTRL and press C.

WHAT IS HAPPENING

As mentioned earlier, a transistor acts like a digital switch, so the white wire controls the base of the transistor to allow the flow of electrons through the transistor to complete the circuit. So when the GPIO pin is on (otherwise known as high) it turns the digital switch on and vice versa.

The 1K resistor acts to lower the required current taken from the Raspberry Pi to allow the transistor to operate.